

Fixing the Root Node: Efficient Tracking and Detection of 3D Human Pose through Local Solutions

Ben Daubney, Xianghua Xie^{*}, Jingjing Deng

Department of Computer Science, Swansea University, Swansea, SA2 8PP, United Kingdom

Neil Mac Parthaláin, Reyer Zwiggelaar

Department of Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, United Kingdom

Abstract

3D human pose estimation is a very difficult task. In this paper we propose that this problem can be more easily solved by first finding the solutions to a set of easier sub-problems. These are to locally estimate pose conditioned on a fixed root node state, which defines the global position and orientation of the person. The global solution can then be found using information extracted during this procedure. This approach has two key benefits: The first is that each local solution can be found by modeling the articulated object as a kinematic chain, which has far less degrees of freedom than alternative models. The second is that by using this approach we can represent, or support, a much larger area of the posterior than is currently possible. This allows far more robust algorithms to be implemented since there is far less pressure to prune the search space to free up computational resources. We apply this approach to two problems: The first is single frame monocular 3D pose estimation, where we propose a method to directly extract 3D pose without first extracting any intermediate 2D representation or being dependent on strong spatial prior models. The second is multi-view 3D tracking where we show that using the above technique results

^{*}The source code of the proposed method can be downloaded from the following address http://cvision.swan.ac.uk/pose_sourcecode.zip.

^{*}Corresponding author

Email address: X.Xie@swansea.ac.uk (Xianghua Xie^{*})

in an approach that is far more robust than current approaches, without relying on strong temporal prior models. In both domains we demonstrate the strength and versatility of the proposed method.

Keywords: 3D Pose Estimation, Tracking, Local Solutions, Root node.

1. Introduction

There is currently much interest in tracking and estimating the pose of articulated objects, particularly if this articulated object is a human. The principal difficulty with this task is the high dimensionality of the solution space and
5 noisy, often ambiguous observations. The high dimensionality of the solution space can be overcome by, for example, using iterative approaches that allow a coarse to fine resolution search to be performed. Whilst effective at finding strong minima, these approaches are susceptible to ambiguous observations; only a small area of the posterior is supported making these methods prone
10 to unrecoverable failure. This is in contrast to methods such as the Pictorial Structures Model (PSM) [1, 2, 3], where the entire search space is evaluated over a discrete grid. By calculating the full posterior distribution ambiguities can be overcome as either more observations become available or by integrating further high-level *a priori* information. Whilst the PSM is applicable to 2D
15 pose estimation, when applied to 3D pose estimation the search space becomes too large to exhaustively search without resorting to an unsatisfactorily coarse grid.

The approach we propose in this work is to combine the advantages of both methods. On the one hand we want to provide much wider support across the
20 solution space, whilst on the other hand keep computational costs low by actively pruning areas of little interest to enable finer resolution searches to be performed over a continuous, rather than discrete, state space. To accommodate these seemingly conflicting ideas, instead of estimating pose as a single optimization problem, we attempt to solve a set of easier, more constrained intermediate
25 problems. By allowing each intermediate problem to be solved independently, we

can ensure that these are broadly distributed across the domain, thus increasing support. Whilst at a local level to each sub-problem, we can more confidently reduce the search space to provide a set of local solutions.

Each local solution is provided by estimating the conditional posterior distribution. This is a distribution over human pose conditioned on a fixed state of the root node. The root node represents the position and orientation of the person in the global frame of reference. The principal assumption we exploit in this work is that *it is much easier to correctly estimate 3D pose if the correct state of the root node is known a priori*. To exploit this assumption we only require that: **1. The correct state of the root node is contained within the solution set. 2: Given (1), that the correct root node state can then be identified.** The first condition is met by ensuring local solutions are found for a broad range of root node states. The second condition is achieved by using knowledge gained through estimating each local solution, for example, by finding the local solution with the highest likelihood.

However, the technical challenge in using this methodology is not to allow the computational cost of finding multiple solutions to be greater than competing methods that estimate pose as a single optimization problem. Through this work we show this can be achieved and present a number of novel approaches to attain this whilst applying a local-solution approach to two different problems. The first is direct 3D pose estimation from single monocular images and the second is 3D multiple-view tracking of pose. It is shown that in both scenarios state-of-the-art results are achieved without providing our method with additional computational resources, where we assume the computational bottleneck is in the required number of image likelihood evaluations.

2. Background

Human pose estimation can be broadly split into two categories, detection and tracking. The key difference between the two is in the source of prior information used by each. In tracking this information is provided through ob-

55 servations made in the previous time steps and a temporal prior that describes
how a part is expected to move. For pose detection it is provided by a spatial
prior that describes the relationships between connected parts. The prior effec-
tively adds a set of constraints. In general the more constrained the prior the
better the method will work given noisy data, though this is at the expense of
60 its generality.

Simple and unconstrained priors include zero-mean Gaussian diffusion for
tracking [4] and a uniform prior for pose detection [5]. However, the focus of
much recent work has been on developing stronger priors. For tracking, action
specific models are learned using methods such as Gaussian Process Dynam-
65 ical Models (GPDM) [6] or Mixture of Factor Analyzers [7]. These methods
effectively reduce the dimensionality of the pose space by exploiting repeated
patterns of motion in actions such as walking or running. A benefit is that
they can learn correlations between unconnected parts of the body allowing a
part to be localized even if it is occluded, though this is at the expense that
70 this approach will deteriorate for unseen motions or poses. This limitation can
be overcome by learning a range of priors for different motions and extracting
the required prior at runtime [8]. In this work for tracking we use a zero mean
diffusion model as a temporal prior. We opt for this weak temporal prior since
this exposes the performance of the underlying tracking methods in coping with
75 any noisy or ambiguous observations, not the strength of the prior model or
data it has been learned from.

In pose detection it is often desirable to keep the model as general as pos-
sible so it is applicable to a variety of poses. For example a single Gaussian
may represent the prior between connected parts [9]. Correlations between un-
80 connected parts are modeled by adding latent variables [10] or learning a set
of more constrained individual priors by first clustering training data and then
learning a model from each cluster [11, 12].

Whilst still an open problem, some have attempted to combine tracking and
detection using both strong temporal and spatial priors [13, 5, 14, 15, 16, 17], and
85 very recently coupling action and pose estimation [18], [19]. However, directly

combining the two often results in an untractable optimization problem where the global solution can not be guaranteed [13]. A more popular method is to effectively treat the two as independent problems [5, 16, 14]. Temporal priors are used to reduce occurrences of false positives, whilst improving true positive rates. A further benefit is that temporal consistency of the appearance of parts across a sequence can also be exploited.

In addition to a prior, another key component needed for estimating human pose is a method of optimization or inference. Currently, for single image 2D pose estimation a popular method is the Pictorial Structures Model (PSM) [9, 20, 21]. This is a part based approach, where each part is detected independently and then these detections are assembled into the most likely configuration using a spatial prior and Dynamic Programming. This approach assumes a tree structure, where nodes represent the parts of the model and physically connected parts are joined by edges. The search space for each part is defined by a uniformly sampled grid that covers all permissible orientations and positions. The benefit of a uniformly sample grid is that the maximum coverage of the search space is achieved given the available resources, there is no bias as a result of initialization. Additional edges can be added to the model to represent temporal connections, however, often the problem then becomes intractable and methods such as Loopy Belief Propagation [22] or using a combination of trees [15] can be used to find a local solution. Recently, Deep Neural Networks (DNNs) show outstanding performance on different vision problems, such as large scale visual recognition and object detection. DeepPose method [23] treats the pose estimation as a regression problem, where it learns the correlation between the pose vector (coordinates of the joints in the bounding box of detected subject) and the image appearance using convolutional DNN. It requires a large amount of training data, and hard to extend to 3D, as there are much more degrees of freedom than 2D image domain.

Whilst popular for 2D pose estimation it is not obvious how to apply these uniformly sampled grid approaches to 3D pose estimation. The main difficulty is how to discretize the search space of a more complex and higher dimensional

object and negate the additional computation cost of exploring this space. For this reason stochastic approaches are popular for 3D pose estimation [4, 24, 25, 13, 26, 27]. Each stochastic sample may represent the entire state of the body [4, 27, 25, 24] or an individual part [13, 26, 28]. Intuitively, estimating the entire state is more computationally intensive since the size of the search space is exponential with the number of parts, though methods have been developed to improve the efficiency of this task [4, 29]. An alternative is to optimize individual components of the object’s state, for example using Partitioned Sampling [25] or Markov Chain Monte Carlo [24]. A limitation with these approaches is that they are iterative and need convergence for a solution to be found. As noted in [30, 31], this convergence happens in a particular order for objects modeled as a kinematic chain. Typically, those parts nearer a fixed node must converge before parts further down the model can do so. It is expected that any uncertainty in a given part will be propagated down the kinematic chain.

To overcome this problem, stochastic part based methods can be used, such as Non-Parametric Belief Propagation [32, 13] or Variational MAP [26]. These approaches do not model an articulated model as a kinematic chain but as a loose-limbed model, where the joint between connected parts is soft and allowed to deform. However, as the connection between parts is soft, the model is less constrained and slippage can occur, where two limbs can be joined at a very unlikely location or may not even be physically joined. It has been shown that given a known root location, models that have fixed joint positions outperform loose-limbed models at estimating 3D pose [31].

A popular alternative to direct 3D pose estimation is to first estimate 2D pose and then “lift” this to 3D using a low dimensional embedding of the action you are observing [16, 6, 33]. However, the limitation of this is that whilst the 2D prior is likely to be very general the mapping between 2D and 3D will most likely not be. Other approaches include estimating pose from multiple visual hulls [34] and from dept images [35, 36] since the recent growing popularity of time-of-flight sensors.

The method we propose in this work allows the benefits of a part based ap-

proach to be exploited, whilst still modeling the body as a kinematic chain. This results in a method that is constrained, allowing accurate pose estimation to be performed, yet efficient. This is achieved by fixing the root node state for each local solution and finding the conditional posterior for each. All probability density functions are represented by parametric models making the representation extremely efficient compared to purely particle based approaches.

The approach developed in this work is applied to two problems, monocular 3D pose estimation and multi-view 3D pose tracking. The tracking method has been previously published in a conference proceedings [37], however, in this paper we significantly strengthen the principal and theoretical grounding for the approach. This permits us to develop a much more general framework and we demonstrate this by also applying it to the problem of unconstrained monocular pose estimation. The tracking method described can be seen as a single implementation, or incarnation, of the framework described herein.

3. Approach

To estimate human pose we use a part based approach. The body is represented as a graph consisting of n hidden and n observable nodes. The hidden nodes represent the state of different parts of the kinematic chain, $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, and the observable nodes represent the observation for each part, $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$. Each observable node is connected to a single hidden node and hidden nodes are connected by the set of edges $(v_i, v_j) \in E$. In this work we place particular emphasis on the root node, \mathbf{x}_r . The state of this node is particularly important since it also gives the pose its global configuration, for example its position and orientation. The state of all other nodes describe the local orientation and position of each part relative to this. We therefore can decompose the hidden states as $\mathbf{X} = (X, \mathbf{x}_r)$, where $X = \{\mathbf{x}_1, \dots, \mathbf{x}_{n-1}\}$ represents all nodes excluding the root node. A graphical representation can be found in Table 1, where only the hidden nodes are shown.

We use two steps to estimate pose. The first step is to find a set of local

solutions $\{(X^{1*}, \mathbf{x}_r^1), \dots, (X^{l*}, \mathbf{x}_r^l)\}$, where each local solution is given by the pose that maximizes the conditional distribution

$$X^{m*} = \underset{X}{\operatorname{argmax}} p(X|Z, \mathbf{x}_r^m). \quad (1)$$

Given the set of local solutions, or pose estimates, we then seek to find the “best” local solution. A number of measures could be used to achieve this for example, by finding the pose that maximizes the posterior or learn discriminative detectors to locate the correct local solution.

180 This method is in direct contrast to a hierarchical approach where it attempts to first accurately locate the global parameters (position, orientation), following which it then estimates the pose. In this work, we first accurately estimate the pose and then the location. The benefit of this approach is that it can use information extracted during pose estimation to provide a more confident
185 estimate of the global parameters (i.e. root node state.).

3.1. Probability Density Function Representation

Probability Density Functions (PDFs) are commonly represented using discrete samples. The distribution or density of the samples may represent the PDF if using a method such as the Particle Filter [27] or, alternatively the
190 samples may be taken uniformly over a grid and weighted by the PDF at that position. This is particularly popular if using a method such as Dynamic Programming or Belief Propagation [9]. We refer to these samples as delta-samples since they only represent the PDF at a single position. In the case of Kernel Density Estimation (KDE), each sample is assumed not to be discrete but have
195 a continuous distribution defined by the parameters of the Kernel.

In this work, we represent the PDF using a set of hyper-samples, these are a fusion of both delta-samples and a parametric representation. Each hyper-sample only provides support at a discrete location in the state space of the root node, however, the distribution over all other parts is continuous and represented using a parametric model. A hyper-sample is therefore defined as $S^m = \{\mathbf{x}_r^m, \Theta^m\}$, where \mathbf{x}_r^m is the delta-sample representing the root node state

and Θ^m is a parameter to describe the PDF over all other parts. In addition each hyper-sample is provided a weight where

$$w^m \propto p(\mathbf{x}_r^m) \quad (2)$$

which is the prior of the root node state.

The PDF over pose \mathbf{X} is approximated by a set of M hyper-samples

$$p(\mathbf{X}) \approx [S^m]_{m=1}^M, \quad (3)$$

where each hyper-sample represents the PDF over all parts conditioned on a given root node state:

$$S^m = p(X|\mathbf{x}_r^m). \quad (4)$$

Each hyper-sample can further be decomposed to a distribution over each part, excluding the root node,

$$p(X|\mathbf{x}_r^m) = \{p(\mathbf{x}_1|\mathbf{x}_r^m), \dots, p(\mathbf{x}_{n-1}|\mathbf{x}_r^m)\}. \quad (5)$$

Given a set of hyper-samples $\{S^1, \dots, S^l\}$, the probability for a given configuration is then calculated as

$$p(\mathbf{X}) = \sum_{m=1}^l w^m p(X|\mathbf{x}_r^m) \delta(\mathbf{x}_r^m - \mathbf{x}_r), \quad (6)$$

where $\delta(\cdot)$ is the Dirac delta function and $\sum_{m=1}^l w^m = 1$. The conditional probability is given by the parametric function

$$p(X|\mathbf{x}_r^m) = F(X, \mathbf{x}_r^m, \Theta^m), \quad (7)$$

Whilst $F(X, \mathbf{x}_r^m, \Theta^m)$ could be represented by any suitable function that could be used to represent a probability density function, in this work we examine using a graphical representation. A star graphical model is used for pose estimation so that

$$p(X|\mathbf{x}_r^m) = \prod_{i=1}^{n-1} p(\mathbf{x}_i|\mathbf{x}_r^m) \quad (8)$$

where

$$p(\mathbf{x}_i|\mathbf{x}_r^m) = f(\mathbf{x}_i, \mathbf{x}_r^m, \Theta_i^m) \quad (9)$$

and the parameter for each hyper-sample becomes a set, $\Theta^{\mathbf{m}} = \{\Theta_1^m, \dots, \Theta_{n-1}^m\}$. For tracking we use a model based on the standard Pictorial Structures model [9] assuming that the graph is a tree and does not contain any loops

$$p(X|\mathbf{x}_r^m) = \prod_{i=1}^{n-1} p(\mathbf{x}_i|\mathbf{x}_r^m) \prod_{(v_i, v_j) \in E} p(\mathbf{x}_i|\mathbf{x}_j, \theta_{ij}) \quad (10)$$

where θ_{ij} is a connection parameter which describes how probable a configuration is between two connected parts, and (v_i, v_j) are a pair of adjacent nodes in the structure model. Unlike $\Theta^{\mathbf{m}} = \{\Theta_1^m, \dots, \Theta_{n-1}^m\}$, these connection parameters are constant across all hyper-samples. $\prod p(\mathbf{x}_i|\mathbf{x}_j, \theta_{ij})$ represents the conditional dependence between connected parts where the connection graph is illustrated in Table 1. Note also that as \mathbf{x}_r^m remains constant when estimating a local solution, for each hyper-sample the addition of a dependence on this value in Eqn. (10) does not introduce loops into the graph.

Each hyper-sample S^m represents a hyper-plane of the PDF over all parts, except the root node. Each hyper-plane is parallel to the axes of the root node state and passes through the point \mathbf{x}_r^m . This method of approximating a PDF is illustrated in Figure 1 and compared to using a set of delta-samples. As can be seen a small set of hyper-samples can represent a large area of the PDF and are much more informative. For example each mode is easily accessible through the parameters of S^m , whereas further analysis, such as clustering, would need to be applied to extract the modes of the representation depicted in Figure 1 (a). This representation differs from KDE since we do not combine the distributions from different hyper-samples. Each hyper-sample independently represents a slice of the probability density function over the state space \mathbf{X} .

3.2. Estimating a Local Solution

To estimate a local solution the observational likelihood function must also be computed. Using the standard Pictorial Structures model and Bayes' theorem, the probability of a configuration given a fixed root node value and set of

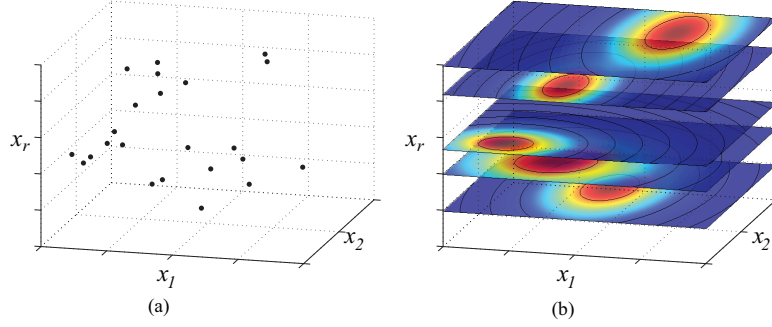


Figure 1: Comparison of representing a PDF using a set of delta-samples (a) versus Hyper-Samples (b). Whilst in standard approaches a sample typically represents the PDF at a single location (a), using the proposed method each hyper-sample represents a hyperplane of the PDF conditioned on the root node state, \mathbf{x}_r (b).

observations is calculated as

$$p(X|Z, \mathbf{x}_r^m) = \frac{p(Z|X)p(X|\mathbf{x}_r^m)}{p(Z)} \quad (11)$$

where the likelihood for each node is independent so that

$$p(Z|X) = \prod_{i=1}^{n-1} p(\mathbf{z}_i|\mathbf{x}_i). \quad (12)$$

The distribution $p(X|\mathbf{x}_r^m)$ is calculated using Eqns. (8) or (10). To estimate a local solution we use Belief Propagation to calculate the belief at each node given by

$$p(\mathbf{x}_i|Z, \mathbf{x}_r) = p(\mathbf{z}_i|\mathbf{x}_i)p(\mathbf{x}_i|\mathbf{x}_r^m) \prod_{v_j \in \mathcal{E}(i)} p(\mathbf{x}_i|\mathbf{z}_j, \dots, \mathbf{z}_T), \quad (13)$$

where $v_j \in \mathcal{E}(i)$ defines the set of edges connected to i and $\{\mathbf{z}_j, \dots, \mathbf{z}_T\}$ represents the set of observations for the subtree containing v_j , created by removing the edge $\{v_i, v_j\}$. The right most term represents messages being passed from connected nodes.

The beliefs are calculated using Importance Sampling where delta-samples are drawn from the proposal function given by

$$\mathbf{x}_i^l \sim p(\mathbf{x}_i|\mathbf{x}_r^m) \prod_{v_j \in \mathcal{E}(i)} p(\mathbf{x}_i|\mathbf{z}_j, \dots, \mathbf{z}_T) \quad (14)$$

and weighted by the likelihood function $w_i^l \propto p(z_i|x_i^l)$. Using these weights the parameters of the hyper-samples $\Theta^m = \{\Theta_1^m, \dots, \Theta_{n-1}^m\}$ can then be updated so that the distribution $p(X|Z, \mathbf{x}_r^m) \approx F(X, \mathbf{x}_r^m, \Theta^m)$. The method used to draw delta-samples and calculate the proposal function is different for both pose estimation and human tracking, as is the method to update the hyper-sample parameters.

In the following sections, we use hyper-samples to propose new novel methods for both 3D pose estimation from single images, and 3D tracking where multiple views are available. Whilst exploiting the same framework the two approaches are implemented very differently to highlight the strength of this approach. For tracking, the hyper-samples are distributed over \mathbf{x}_r stochastically, similar to existing particle filtering approaches, though in contrast the delta-samples are selected deterministically making the approach extremely efficient. For single frame pose estimation the hyper-samples are distributed uniformly over \mathbf{x}_r as in the PSM model to ensure maximum coverage of the pose state space, however, now the delta-samples are drawn stochastically. For monocular pose estimation we use a discriminative likelihood function, $p(\mathbf{z}_i|\mathbf{x}_i)$, whilst for tracking it is generative. Furthermore, to estimate the correct root node state, \mathbf{x}_r^* , a discriminatively trained detector is used for monocular pose estimation, whereas for tracking we use a generative model. The PDF over each individual part, $p(\mathbf{x}_i|\mathbf{x}_r^m) = f(\mathbf{x}_i, \Theta_i^m)$, is modeled using a single gaussian for tracking and a Gaussian Mixture Model for single frame monocular pose estimation.

In order to demonstrate that for optimizing over articulated objects, a good global solution can be found by first finding a set of local solutions, and then optimising over these to find the best global solution, we apply the proposed method to both pose estimation and pose tracking problems. The graphical model used to represent the human body for each task is shown in Table 1 (top row). The node labels for Monocular Pose estimation are Torso, Head (H), Left Arm (LA), Left Leg (LL) etc. The labels for Multi-View Tracking correspond to the Hip (Hip), Torso (Tor), Upper Left Arm (ULA), Lower Left Arm (LLA), Upper Left Leg (ULL) etc. There is a hidden node for each observed node, so

Table 1: Comparison of method presented for monocular pose estimation and multi-view tracking. Root nodes are shaded and for clarity observable nodes are not shown.

	Monocular 3D Pose	Multi-View Tracking
Graph		
Estimate $p(X Z, \mathbf{x}_r^m)$	Stochastic	Deterministic
Hyper-sample distribution over \mathbf{x}_r	Deterministic	Stochastic
Estimate \mathbf{x}_r^*	Discriminative	Generative
Estimate $p(\mathbf{z}_i \mathbf{x}_i)$	Discriminative	Generative
Parameterization of $p(\mathbf{x}_i \mathbf{x}_r^m)$	Gaussian Mixture Model (GMM)	Gaussian

it is one to one mapping as in the standard pictorial structure mode. As can be seen, for monocular pose estimation a single node is used to represent an entire limb, whereas for tracking a node defines a single part. A comparison of the two solutions can be seen in Table 1, and a full description of the model used for each is provided in the following sections.

4. Application 1: Monocular 3D Pose Detection

In this section, we apply our method to the problem of estimating 3D pose from single monocular images. The pseudocode is illustrated in Algorithm 1. Human pose estimation benefits from a fixed root node approach in a number of ways. Firstly, using our method we ensure maximum coverage of the search space is achieved given a fixed set of resources, by distributing the hyper-samples uniformly across the state space of the root node. Secondly, we use limb likelihood estimates for a given local solution to train discriminative human detectors to improve detection rates. Finally, using our method, an accurate solution can be located without the requirement of convergence as the hyper-samples can

easily represent multiple modes, even when conditioned on a single root node value. This is as the prior is modeled using a Gaussian Mixture Model (GMM) and we show that each component, learned in quaternion space, represents an independent volume when projected into Euclidean space.

Algorithm 1 Algorithm for Monocular 3D Pose Detection

Given a set of hyper-samples uniformly distributed over the root node parameter space and initialised to the model prior.

for each hyper sample **do**

 Stochastically optimize to find $p(X^*|x_r^m)$.

end for

Find $p(X^*)$ given by the hyper sample which returns a positive detection from the detector.

Further refine $p(X^*|x_r^m)$ using more expensive image features.

4.1. Model Representation and Sampling from the Prior

The graphical model used in this section is a star, consisting of 6 nodes; the root node, which represents the torso and 5 nodes representing each of the main limbs (heads, arms and legs). We assume the position of the ground plane is known, which is a common assumption for 3D pose estimation and tracking [14, 38], though methods do exist that could be used to automate this process (e.g. [39]). The state of the root node is parameterized as $\mathbf{x}_r = (\mathbf{d}_r, q_r)$, where $\mathbf{d}_r \in \mathbb{R}^2$ defines the position on the ground plane and $\{q_r \in \mathbb{R}, 0 \leq q_r < 2\pi\}$ defines the heading of the subject.

Each distribution $p(\mathbf{x}_i|\mathbf{x}_r^m)$ is modeled using a GMM. For each limb, a GMM is learned and used to initialize each hyper-sample respectively, hence $p(\mathbf{x}_i|\mathbf{x}_r^m) = \sum_{k=1}^K \lambda^k \mathcal{N}(\mathbf{x}_i; \mu_i^k, \Sigma_i^k)$, so that $\Theta_i^m = \{\lambda_i^k, \mu_i^k, \Sigma_i^k\}_{k=1}^K$, where K is the number of components in the model, and λ_k, μ_k and Σ_k represent the k th component's weight, mean and covariance respectively. The root node state is taken from a three dimensional grid over \mathbf{x}_r , representing locations on the ground plane and at each position a set of discrete orientations. This is de-

picted in Figure 2 (c) where we visualize a subset of the initial hyper-samples used to estimate pose. In practice this is sampled much more densely and at each sample point there are also hyper-samples with different orientations. Each hyper-sample is therefore parameterized as $S^m = \{\mathbf{x}_r^m, \Theta_1^m, \dots, \Theta_{n-1}^m, w^m\}$,
 290 where $\Theta_i^m = \{\lambda_i^k, \mu_i^k, \Sigma_i^k\}_{k=1}^K$.

Each distribution is learned over possible limb rotations, which are represented as unit quaternions. To approximate a Gaussian distribution over quaternions we use an approach similar to [13]. Each unit quaternion is represented
 295 by two parts a scalar and vector part $\mathbf{q} = q_0 + \bar{\mathbf{q}}$. By ensuring the scalar component is positive a quaternion can be represented in \mathbb{R}^3 using only the vector part. To reduce the likelihood of training data being located across the edge of the unit sphere, the training data is used to estimate a “safe” quaternion space by rotating the data so that the sum of the scalar component across all data is
 300 maximal [37]. A GMM can then be learned directly in this space for each limb independently.

Each model is learned over an entire limb, which is represented by a single node. This may represent a distribution over more than a single part, i.e. a distribution over the left leg models that over both the lower and upper leg,
 305 hence $\mathbf{x}_i \in \mathbb{R}^6$ (since each part has three degrees of freedom), except for the head which is modeled as a single part, $\mathbf{x}_{head} \in \mathbb{R}^3$. The covariance for each part is diagonal and can be written as $\Sigma_i^k = \text{diag}(\mathbf{x}_{i1}^k, \dots, \mathbf{x}_{ij}^k, \dots, \mathbf{x}_{ip}^k)$, where j is the index of the part, p is the number of parts for a given limb. For the arms and legs $p = 2$, and for the head $p = 1$. The rotations are defined in the frame of
 310 reference of the root node, not the part to which they are physically connected.

As the graphical model is a star with a fixed root node the proposal function defined in Eqn. (14) becomes $p(\mathbf{x}_i | \mathbf{x}_r^m)$. Delta-samples are drawn from each GMM by first picking a component with likelihood $k^* \propto \lambda_k$, following which a sample is drawn from the selected component $(\mathbf{x}_{i1}^s, \dots, \mathbf{x}_{ip}^s)^T \sim$
 315 $\mathcal{N}((\mathbf{x}_{i1}, \dots, \mathbf{x}_{ip})^T; \mu_{ij}^{k*}, \Sigma_{ij}^{k*})$, where p is the number of parts that make up a given limb. The rotations described by the sample can then be applied to each limb and the kinematic chain is assembled. The root node state, \mathbf{x}_r^m , gives the pose

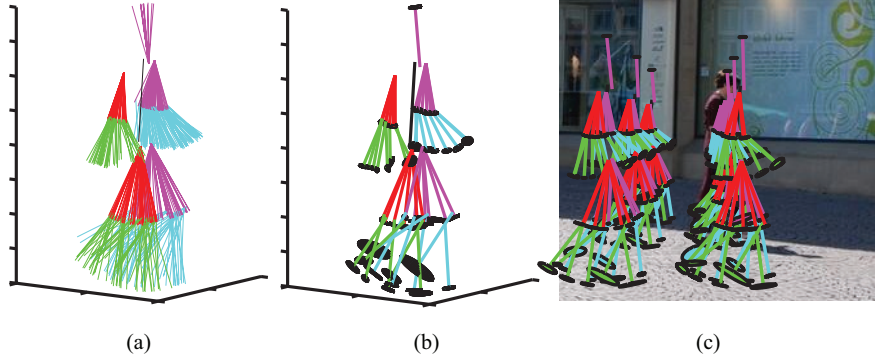


Figure 2: Example of samples drawn from the model prior (a). In (b) the GMM components have been visualized by fitting a covariance to the samples drawn from each. In (c) the model is shown projected to different root positions, though in practice this would be much more dense and at each position multiple orientations would be projected. Lighter colors indicate left side and darker ones indicate right side of the body.

its global position and orientation.

Though a single delta-sample may represent the state of more than one part,
 an observational likelihood model is learned for each part independently. Therefore, the observational likelihood for a given delta-sample is given by combining the likelihood of each part assuming conditional independence, hence $p(\mathbf{z}_i | \mathbf{x}_i^s) = \prod_{j=1}^p p(\mathbf{z}_{ij} | \mathbf{x}_{ij}^s)$. Thus, the weight of each delta-sample is $\pi_i^s = \prod_{j=1}^p \pi_{ij}^s$. The calculated weights are then used to update the GMM components from which they were drawn using the Maximum Likelihood estimate. Note that if all weights were uniform the covariance and mean would be unchanged. The MAP estimate for a local solution is approximated by finding the GMM component with the highest likelihood for each node and using the mean.

An example of the prior used in this work is shown in Figure 2. In (a) we show delta-samples drawn from the prior for a fixed root node position and in (b), for visualization a Gaussian has been fitted in Euclidian space to the delta-samples drawn from each component in quaternion space. As can be seen a different component learned in quaternion space appears to correspond to an independent area in Euclidian space.

335 4.2. Part Detection and Feature Extraction

To detect individual parts we learn discriminative part detectors. The features used are based on the Histogram of Orientated Gradient (HOG) [40], where the gradient magnitudes in a small rectangular region of the image are binned depending on their orientation to form a histogram. A set of these features, which define the detection window, are then concatenated together to form a
340 vector that can be used for training. Each component of this vector is referred to as an *attribute*.

In previous approaches applied to 2D pose estimation, a small number of orientations and scales are searched over for each part (e.g. [41]) allowing the
345 image to be pre-rotated and scaled before feature extraction commences. In our approach a sample for a part could be projected into the image at an arbitrary scale and orientation preventing this method from being applied. To make our feature extraction more efficient we take a number of steps. Firstly, we use the l_1 norm to normalize each feature, which allows us to use an integral
350 image representation to compute histograms efficiently [42] and we compose each feature of just a single cell. To accommodate rotations, instead of orientating individual features we maintain them as squares with the direction of their sides axis aligned with the image, however, we rotate the histogram bins to make them axis aligned in the local frame of reference of the individual part. This is similar
355 to the method applied to orientate SIFT features [43]. To apply scale changes and rotations to the dense grid of individual features that compose a detection window, we directly scale and rotate this grid. This approach can be seen in Figure 3 where we show a set of features projected onto the position of the arm. The scale of each feature is dependent on the hypothesized depth of the subject
360 for which they are being calculated.

The detector used in this work is a JRIP classifier [44]. This is a rule induction approach which learns propositional rules by repeatedly growing rules and then pruning them. During the growth phase, attributes are added greedily until a termination condition is satisfied. These are then pruned in the next
365 phase subject to a pruning metric. Once the rule set is generated, a further

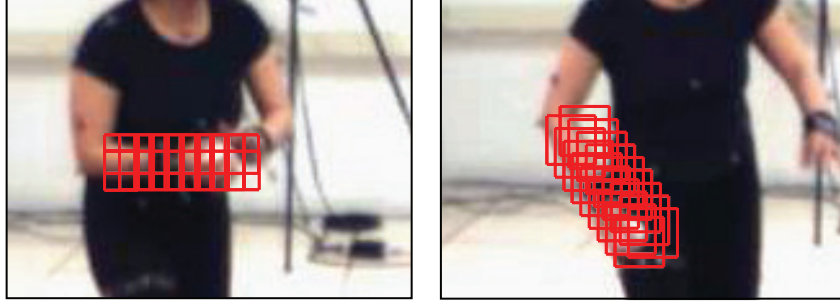


Figure 3: Example showing features projected onto the lower arm. Whilst the grid of the detection window is rotated the corners of each feature are not.

optimization is performed where rules are evaluated and deleted based on their performance on randomized data. A benefit of this approach is that for a given part not all individual attributes are used in the set of rules learned as a result only a small fraction of the features need to be calculated. Classifying a
 370 detector window is thus extremely fast. The detector produces a binary decision $\gamma(\mathbf{x}_i, \mathbf{z}_i) = \{true, false\}$. Whilst the JRIP classifier does not explicitly provide a likelihood distribution, we approximate this using detection rates estimated during training. For example, $p(\mathbf{z}_i | \mathbf{x}_i^s, \gamma(\mathbf{x}_i^s, \mathbf{z}_i) = true) \approx \frac{TP_i}{TP_i + FP_i}$. Likewise given a negative detection $p(\mathbf{z}_i | \mathbf{x}_i^s, \gamma(\mathbf{x}_i^s, \mathbf{z}_i) = false) \approx \frac{FN_i}{FN_i + TN_i}$. We also learn
 375 a detector for the torso, though as this is fixed relative to the root node state, only a single delta-sample is generated for this part.

To estimate global orientation and position of a person, \mathbf{x}_r^* , we use a method similar to that presented in [16]. For each hyper-sample, S^m , a feature vector can be constructed using the weights for each delta-sample extracted to update it, $\mathcal{Y}^m = (b_1^m, \dots, b_k^m)$, where k is the number of parts and

$$b_{ij}^m = \frac{1}{d} \sum_{s \in \mathcal{D}_i} \pi_{ij}^s \quad (15)$$

where d is the number of delta-samples and π_{ij}^s is the sample's weight for the j th part of the i th node (i.e. limb). In total there are ten parts, two for each of the main limbs one for the head and one for the torso, hence $\mathcal{Y}^m \in \mathbb{R}^{10}$.

380 As previously described the set of hyper-samples, $S^m \in \mathcal{S}$, are initially

the same except that their root node states, $\mathbf{x}_r^m = (\mathbf{d}_r^m, q_r^m)$, are distributed uniformly across a grid that describes different locations on the ground plane and a set of different orientations. At each ground plane location, \mathbf{d}_{gp} , there are therefore a set of n hyper-samples, $\mathcal{V} = \{S^m | \mathbf{d}_r^m = \mathbf{d}_{gp}\}$, with the same
385 ground plane position, where n is the number of discrete orientations. The feature we use to both detect a person’s position and orientation is constructed by concatenating the likelihoods of all these hyper-samples together. Hence, $\mathbf{V}(\mathcal{V}) = (\mathcal{Y}^1, \dots, \mathcal{Y}^n)^T$. Since we define 16 different orientations, $\mathbf{V}(\mathcal{V}) \in \mathbb{R}^{160}$. These are used as features for training detectors.

390 4.3. Experiments

We use two datasets to train and test our human pose estimation method. The HumanEva dataset [45] provides ground truth motion capture data so that the accuracy of the pose estimation can be quantified and we also use the TUD
395 Multiview Pedestrians Dataset [16] to test our approach on more unconstrained and cluttered scenes.

The position of the ground plane is provided before detection commences. The hyper-samples are distributed uniformly over the ground plane with a spatial resolution of $100mm$ and angular resolution of $1/8\pi$. We see this as being the natural equivalent to 2D approaches that uniformly distribute the samples
400 across the image plane. To estimate the local solution of each hyper-sample 600 delta-samples are drawn, where a single delta sample represents the state of only a single part. Often there are several positive detections where a person is located at small perturbations in position and scale relative to the true location of the person. Therefore, after applying our detector we apply the mean-shift
405 algorithm to cluster the detections.

The same JRIP detectors are used on both data sets. One is learned for each part using data taken over all actions and subjects from the Train partition of the HumanEva dataset [45]. MoCap data was used to select positive examples and negative examples. An average reduction of 93% in the number of attributes
410 is achieved using the JRIP detector. This makes the approach far more efficient

since on average only 7% of the feature needs to be extracted from the image. Each classifier has less than 20 rules and the maximum rule length was just 5 conditions.

To improve the accuracy of the extracted pose we further iterate the proposed method using additional image features. We use a publicly available skin detector [46] to improve the estimate of the hands position. To improve the pose estimate for the lower legs we use a generative foot detector constructed from simple filters. We also use edge features by integrating over the HOG bins that are orientated perpendicular to the edge of the projected part. Note this iterative step is only performed for the selected hyper-sample (i.e. the hyper-sample where the person has been detected), hence, is far more computationally efficient than having to calculate these features for all hyper-samples.

4.3.1. *HumanEva Dataset*

The HumanEva dataset [45] is used to provide quantitative results of the extracted poses. The prior is learned for each subject using motion capture data from the corresponding train partition for that person. Example extracted poses are shown in Figure 4. As can be seen the poses shown closely match those of the subject depicted in each sequence. Quantitative results from the Validate partition of the HumanEva dataset are presented in Table 2. These show the error averaged across all joints of the model. As the method is monocular, we present both the relative and absolute error. The absolute error is dominated by errors in estimating the root node state; often the hardest component to extract is the correct depth. However, even if the depth is underestimated a good representation of pose can still often be extracted resulting in a reduced relative pose error. Also for comparison we present the error when the position of the root node is given, and only the orientation and pose is unknown. As can be seen the relative error increase by only about 20mm when the position is unknown. For comparison we compare our method with [16], who use a 2D Pictorial Structure and then “lift” this to 3D using exemplars, they also have a temporal prior modeled using a hierarchical Gaussian Process Latent Variable model. Tested on

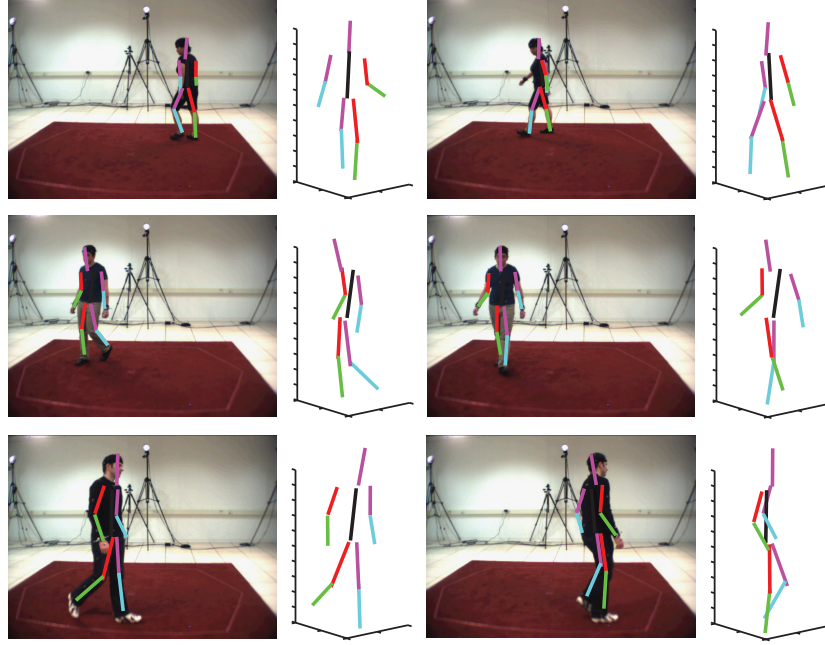


Figure 4: Examples of extracted 3D poses for three different subjects.

walking in a similar sequence they reported a 3D reconstruction error of $104mm$ on one subject and we achieve an error of $104.5mm$ averaged over all subjects. However, their approach has a much stronger prior distribution and they use observations made over multiple frames, where as currently we use only a single
445 image and our prior is much more general.

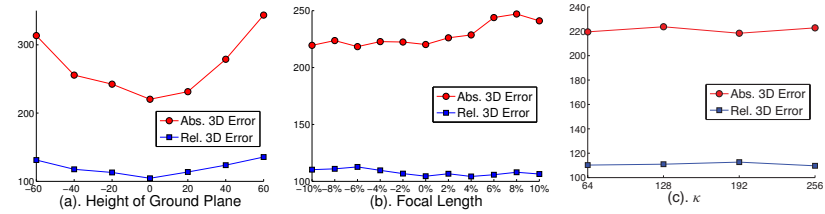


Figure 5: The 3D pose estimation errors for perturbations of ground plane on HumanEva dataset. The units for all Y axis are mm .

To understand the effect of ground plane estimation on pose estimation and tracking, we carried out a study that perturbs the ground plane position

Table 2: Quantitative errors on the HumanEva dataset using Camera 3 [45]. The first column shows the result if the position of the torso is known, but the pose and correct orientation is unknown. The second column shows the relative error and the third absolute. The unit for all quantities is *mm*.

Subject	Known root position	Rel. error	Abs. error
S1	77.7	106.1	256.9
S2	62.1	83.3	200.1
S3	110.3	124.2	203.7
Average	83.3	104.5	238.8

and orientation and focal length. Figure 5(a) shows that when we deviate the height of ground plane by 60mm (+/-, up and down respectively) from the ground truth, an increase of absolute error can be observed, up to 343.53mm error compared to 220.24mm. This is expected as the height of ground plane has a noticeable impact on the position of the root node. Figure 5(b) shows the result of changing the focal length of the camera, which had limited impact on the performance. Rotating the ground plane had little influence on the results, see Figure 5(c). The rotational random perturbation is achieved by sampling from the Von Mises-Fisher distribution in R^3 , with concentration parameter κ controls the degree of perturbation.

4.3.2. Datasets with unconstrained Scenes

The TUD Multiview Pedestrian Dataset contains images of people walking in unconstrained scenes, such as shopping malls or parks. The scenes are highly cluttered and the people differ in both their appearance, location and size.

For training we labeled the position of the person in each image and also classify their orientation to one of sixteen different orientations. This was performed for 248 images used as a training set. The spatial model used in this experiment is learned from Subject 1 in the HumanEva dataset whilst walking. The GMM for each limb conditional is learned from training data using the Expectation Maximization algorithm.

We trained a multi-class linear SVM to estimate the orientation. Using 10 fold cross validation we achieve an accuracy of 0.21, significantly better than

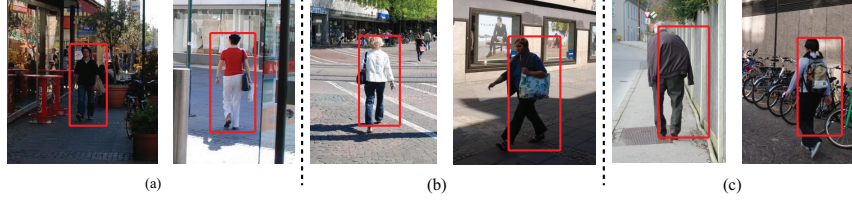


Figure 6: Example frames showing detector bounding box and ranking. (a) Examples with ranking $R1$. (b) Examples with ranking $R2$. (c) Examples with ranking $R3$.

chance (0.0625). Note, commonly for pose estimation, we impose a much stricter criteria for positive detection compared to object detection. This means lower accuracy of the classifier will be reported in order to retain only reliable positive samples. If we relax the criteria slightly, i.e. in this case the discretized orientation state, the overall classification rate will increase significantly. For example, if we allow a tolerance of ± 1 orientation state, an accuracy of 0.44 is achieved.

For testing we use 50 images taken across a broad range of scenes and orientations. To give an idea of our detectors performance we rank detector performance based on scale and position. We rank a detection as $R1$ if the estimated height, H_{est} , in the image plane is within $\pm 15\%$ of the true projected height, H_{GT} , and the root position is estimated within $\pm 0.05 \times H_{GT}$ of the correct root position. A rank $R2$ score indicates the scale was correct by $\pm 30\%$ and root position by $\pm 0.10 \times H_{GT}$. Rank $R3$ is anything worse than this. Over the 50 test images the ranking scores are 24, 12 and 14 for $R1$, $R2$ and $R3$, respectively. Example frames showing the detector’s performance for each are presented in Figure 6. The conditions we use to rank the detections are quite stringent since a small error in the estimated position of the root node can make a significant difference in the resultant pose.

In Figure 7 we show some examples of the extracted pose. In (a), (b) and (d) we visualize the posterior distribution for the optimal hyper-sample, $p(X|Z, \mathbf{z}_r^{m*})$, by extracting delta-samples ((a) and (d)) and by plotting the GMM components (b). As is clearly shown the resultant posterior is still highly multimodal allowing the opportunity for further optimization based on higher

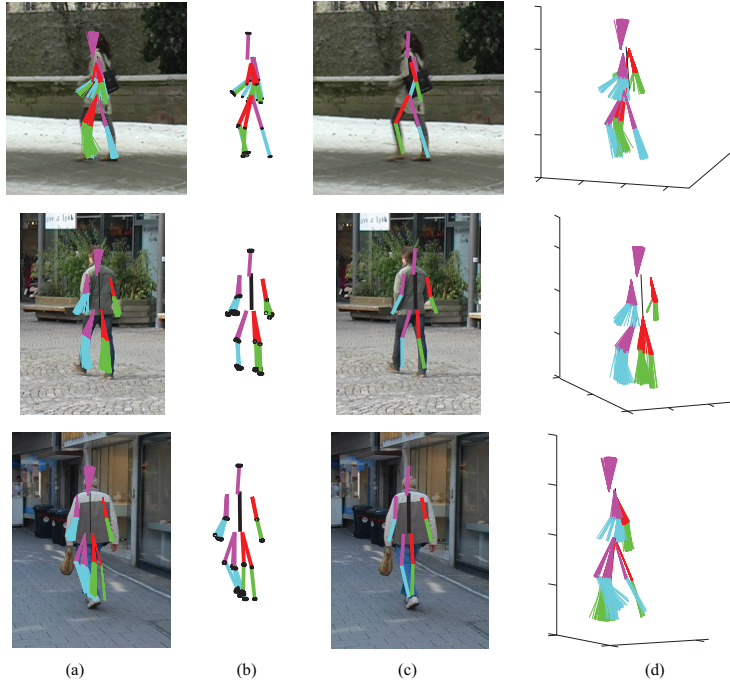


Figure 7: Examples of estimated pose and conditional posterior distribution on TUD Multi-view Pedestrian dataset, $p(X|Z, \mathbf{z}_r^{m*})$. (a) Projection of delta-samples drawn from conditional posterior distribution. (b) Visualization of GMM modes. (c) MAP estimate of pose. (d) Visualizing delta-samples from alternative view.

level priors or temporal integration. In particular notice in the example in the top row that when the samples are rotated slightly as shown in (d), it can be
495 seen that the front leg in the image is represented by both a mode for the left leg and the right. Note also that although we only visualize the distribution for the most likely hyper-sample, all other hyper-samples for all positions and orientations are still maintained and can be accessed if needed. In (c) the MAP estimates are shown for each image, as can be seen these closely relate to the
500 images shown.

In Figure 8, we illustrate the most common cause of errors. As the model is represented using a tree in (a) we see two limbs fitting to the same mode. This is a common problem with all tree based methods. Errors shown in (b)-(d) are

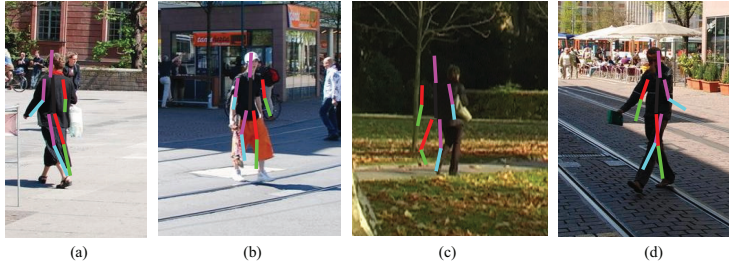


Figure 8: Examples of typical failure cases. (a) Overcounting - both legs are attracted to the same mode. (b) Poor depth estimation. (c) Incorrect root node position estimation. (d) Incorrect root node orientation estimation.

all as a result of incorrect root node estimation, whilst (b) and (c) are due to
505 poor position estimation, (d) shows the wrong orientation has been detected. However, it would be expected that by increasing the resolution of the search over the root node’s position and orientation some of these errors would be reduced.

In addition, we provide examples of 3D pose estimation on Leeds Sports
510 Pose (LSP) dataset [47]. Note, this dataset is designed for 2D pose estimation, i.e. there is no internal camera parameters available or 3D ground truth.

The main parameters concerned are root node state discretization and number of components in GMM for each limb. For the first set of parameters, we only use 16 orientations and the size of each cell on the ground plane is 100×100
515 mm^2 while the ground plane is 2200×2200 . A finer discretization may improve the performance at the expense of some computational cost. The number of components for GMM is empirical set, as in all parametric models. We do not expect this has much influence on the result as long as the number of components is not too small. Generally, the more movements the limb may exhibit,
520 the larger number of components may be needed. We use a single Gaussian for head and torso, 4-components GMM for upper arm, 8 components for lower arm, 3 for upper leg, and 6 GMM for lower leg.

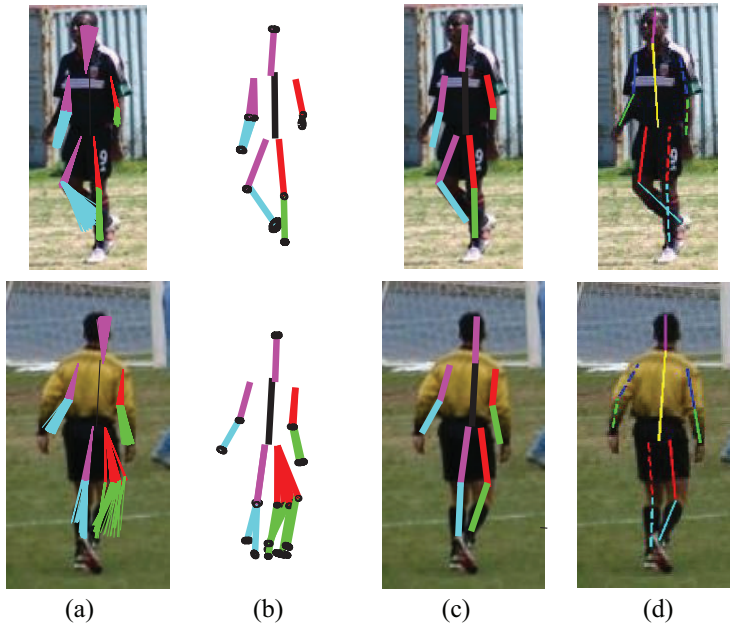


Figure 9: Examples results on LSP dataset. (a) Projection of delta-samples drawn from conditional posterior distribution. (b) Visualization of GMM modes. (c) MAP estimate of pose. (d) 2D ground truth given by manual annotation.

5. Application 2: Multiple View 3D Tracking

In this section, we apply our framework to the problem of tracking a person in 3D using multiple views. The pseudocode is listed in Algorithm 2. The benefit of our approach in this setting is that we can use a set of hyper-samples to represent a much larger volume of the state space than existing methods that typically converge to a single solution or represent very few modes (e.g. [4]). In effect a separate mode is represented by each of our hyper-samples making it a very rich representation. The advantage of this is that it enables our approach to be much more robust to tracking failure. Tracking failure typically occurs when the incorrect mode is tracked. This “incorrectness” is not the fault of the algorithm and does not imply it has failed to track the global maximum. The problem is that observations, even using multiple views, are ambiguous and noisy, therefore it is conceivable that often the global maximum of the posterior is incorrect (i.e. it does not correspond to the true pose). Therefore, if only a single or very few modes are being tracked failure is very likely. Our contribution therefore, is not to design an approach that can most efficiently find the global maximum, which is the focus of much of the tracking literature, but to develop an approach that can support a much larger area of the posterior without further computational cost (i.e. without extra likelihood function evaluations). Therefore, if the global mode is incorrect due to noisy observations the approach is less likely to suffer catastrophic tracking failure. This is achieved by broadly distributing the root node states of the hyper-samples. The effect of this is that it permits greater uncertainty to be represented over the state of the root node, though this is achieved without then propagating this uncertainty to the remaining parts of the model. We show this not to be the case if using existing standard methods, where adding uncertainty to the root node also inflates the uncertainty of all remaining parts of the model.

Further benefits of our approach is that as the PDF over the state space, excluding the root node, is represented parametrically these can be updated in closed-form. For example temporal diffusion across frames can be added by

inflating the covariance of each distribution and messages between connected parts can be computed as a product of Gaussian's. This makes performing inference for each hyper-sample very efficient. Furthermore, we present a method to deterministically extract a sparse set of delta-samples from each distribution. This is motivated by minimizing the KL-divergence between the distribution of the delta-samples and the PDF they are used to approximate. Using this approach each hyper-sample is updated using the equivalent number of image likelihood evaluations as just seven delta-samples in a typical particle filtering approach.

Algorithm 2 Algorithm for Multiple View 3D Tracking

Given a set of hyper samples.

for each hyper sample **do**

Deterministically Optimize to find $p(X^*|x_r^m)$.

end for

Find $p(X^*)$ given by the hyper sample x_r^* with the highest posterior $p(x_r^*)$ as the current solution.

Resample a new set of hyper samples from the old set.

for each new hyper sample **do**

Deterministically inflate the covariances of each hyper sample.

Stochastically perturb the root node state of each hyper sample.

end for

5.1. Tracking and Pose Estimation

Performing a joint optimization over both time and space using a part based approach results in a complex graphical model that is difficult to solve. We take a common approach and assume that tracking can be performed independently to pose estimation and each can be performed in turn.

The model used consists of ten nodes as is common in the Pictorial Structure model, one for each single body part. The state of each part is again represented by a quaternion rotation \mathbf{q}_i that describes the orientation of each part in the

570 frame of reference of the root node. The root node \mathbf{x}_r does not explicitly represent a part, its state represents the position $\mathbf{d}_r \in \mathbb{R}^3$ and orientation $\theta_r \in \mathbb{R}^3$ of the body in the global frame of reference, i.e. that of the motion capture suite.

The PDF for each individual node of a hyper-sample is modeled using a Gaussian distribution, so that $\mathbf{P}(\mathbf{x}_i|\mathbf{x}_r^m) \sim \mathcal{N}(\mathbf{x}_i; \mu_i^m, \Sigma_i^m)$ where μ_i^m and Σ_i^m 575 represent the Gaussian's mean and covariance respectively. Therefore, each hyper-sample is parameterized by $\mathbf{S}^m = \{\mathbf{x}_r^m, \mu_1^m, \Sigma_1^m, \dots, \mu_{n-1}^m, \Sigma_{n-1}^m, w^m\}$.

The posterior at time $t - 1$ is represented by a set of M hyper-samples, so that $p(\mathbf{X}_{t-1}|\mathbf{Z}_{t-1}, \dots, \mathbf{Z}_1) \approx [\mathbf{S}_{t-1}^m]_{m=1}^M$. Temporal propagation of the hyper-samples is performed using importance resampling. A sample, \mathbf{S}_{t-1}^m , is first 580 selected with probability proportional to the sample's weight, w_{t-1}^m . A new sample is then generated from this by propagating it through the temporal model defined as, $p(\mathbf{X}_t|\mathbf{X}_{t-1}) = \{p(\mathbf{x}_{r,t}|\mathbf{x}_{r,t-1}), p(\mathbf{x}_{1,t}|\mathbf{x}_{1,t-1}), \dots, p(\mathbf{x}_{n-1,t}|\mathbf{x}_{n-1,t-1})\}$.

The root node state of the sample is propagated using a Gaussian diffusion model, so that $\mathbf{x}_{r,t}^m = \mathbf{x}_{r,t-1}^m + \mathbf{y}$, where $\mathbf{y} \sim \mathcal{N}(\dot{\mathbf{x}}_r; \dot{\mu}_r, \dot{\Sigma}_r)$ and $\dot{\mathbf{x}}$ represents the 585 first derivative with respect to time.

The remaining parameters of the hyper-sample are propagated using a zero mean diffusion model, however, these can be updated by directly inflating the covariance of each node, so that $\mu_{i,t}^m = \mu_{i,t-1}^m$ and $\Sigma_{i,t}^m = \Sigma_{i,t-1}^m + \dot{\Sigma}_i$, where the covariance, $\dot{\Sigma}_i$, is provided by the temporal prior $p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}) = \mathcal{N}(\dot{\mathbf{x}}_i; \dot{\mu}_i, \dot{\Sigma}_i)$. 590 This temporal prior is not learned directly over $\dot{\mathbf{x}}_i$ but over $\dot{\mathbf{x}}_{ij}$, which is the rotational velocity of the i th part relative to the j th part to which it is connected. Hence, $p(\dot{\mathbf{x}}_{ij}) = \mathcal{N}(\dot{\mathbf{x}}_{ij}, \dot{\mu}_{ij}, \dot{\Sigma}_{ij})$, where this distribution is learned over $\dot{q}_{ij} = (q_{ij}^t)^{-1} q_{ij}^{t+1}$.

Given a value for \mathbf{x}_j this can then be transformed to a distribution over \mathbf{x}_i 595 by:

$$\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1} \sim \mathcal{N}(\mathbf{x}_i, \dot{\mu}_i, \dot{\Sigma}_i) \approx \mathcal{F}\left(\mathbf{x}_j, \mathcal{N}(\dot{\mathbf{x}}_{ij}, \dot{\mu}_{ij}, \dot{\Sigma}_{ij})\right). \quad (16)$$

The transformation, $\mathcal{F}\left(\mathbf{x}_j, \mathcal{N}(\dot{\mathbf{x}}_{ij}, \dot{\mu}_{ij}, \dot{\Sigma}_{ij})\right)$, is non-linear and is performed using the Unscented Transform [48]. This method decomposes the covariance

into a set of $2D$ sigma points $\bar{\Sigma} = \{\sigma_1, \dots, \sigma_{2D}\}$, where D is the dimension of the covariance. Each sigma point is then translated by the mean to generate a set of points that represent the mean and covariance of the original distribution. Each sigma point is calculated as

$$\begin{aligned}\sigma_d &= \mu + \sqrt{Dv_d}\mathbf{e}_d, \\ \sigma_{D+d} &= \mu - \sqrt{Dv_d}\mathbf{e}_d,\end{aligned}\tag{17}$$

where v_d and \mathbf{e}_d represents the d th eigenvalue and eigenvector of the covariance matrix. Each sigma point is then propagated through the non-linear function (i.e. $\sigma'_m = \mathbf{q}_i\sigma_m$) and the mean and covariance calculated from them.

During resampling, methods from annealing are used to adjust the weight of the hyper-samples, such that $w' = (w)^\beta$. A value of β is selected such that the particle survival rate α can be estimated over the entire set of hyper-samples as described in [4]. To allow the same survival rate to be maintained over a fixed time interval, α is set according to $\alpha = \exp \frac{\ln \alpha_c}{N_t}$, where α_c is the desired cumulative survival rate per second and N_t is the frame rate. This is used so that the uncertainty over the root node can be consistent regardless of the frame rate. A larger value of α_c will allow the distribution of the hyper-samples to spread over a larger area of the root node state space, since more of the sample population will be maintained. This will provide wider support of the posterior distribution.

5.2. Local Solution Estimation

In this section we describe how a single hyper-sample is optimized to find a local solution. Since we assume this can be performed independent of time we drop the temporal indices for brevity. Though note the process described must be performed for each hyper-sample in turn.

5.2.1. Limb Conditionals

Limb conditionals describe how two connected parts can deform relative to one another and are described by the distribution $p(\mathbf{x}_j|\mathbf{x}_i, \theta_{ij})$, where θ_{ij} is

the connection parameter. Rather than learning a full limb conditional over \mathbf{x}_i and \mathbf{x}_j we follow the approximation in [13] and learn a distribution over \mathbf{x}_{ij} (i.e. $p(\mathbf{x}_{ij}|\theta_{ij})$). This distribution is also learned over unit quaternions, where $\mathbf{q}_{ij} = \mathbf{q}_i^{-1}\mathbf{q}_j$, and the connection parameters are defined as the mean, μ_{ij} , and covariance, Σ_{ij} , of a Gaussian distribution. Given a state for \mathbf{x}_i a PDF over \mathbf{x}_j can be estimated by propagating the distribution, $\mathcal{N}(\mathbf{x}_{ij}; \mu_{ij}, \Sigma_{ij})$, through the rotation \mathbf{q}_i . This is also performed using the Unscented Transform, used in the previous section, and is described by

$$p(\mathbf{x}_j|\mathbf{x}_i, \theta_{ij}) \approx \mathcal{F}(\mathbf{q}_i, \mathcal{N}(\mathbf{x}_{ij}; \mu_{ij}, \Sigma_{ij})). \quad (18)$$

5.2.2. Calculating Beliefs

In this section, we describe how the states of the nodes are updated for each hyper-sample S^m using message passing between nodes. The messages are calculated using Importance Sampling by drawing delta-samples from the proposal distribution

$$\mathbf{x}_j^l \sim p(\mathbf{x}_j|\mathbf{x}_r^m) \prod_{v_k \in \mathcal{E}(j)} p(\mathbf{x}_j|\mathbf{z}_k, \dots, \mathbf{z}_T). \quad (19)$$

By constraining that all messages are Gaussian the proposal function is itself a Gaussian distribution with mean and covariance

$$\begin{aligned} \Sigma_j^{-1} &= (\Sigma_j^m)^{-1} + \sum_{v_k \in \mathcal{E}(j)} \left(\Sigma_k^{\vec{k}j} \right)^{-1} \\ \Sigma_j^{-1} \mu_j &= (\Sigma_j^m)^{-1} \mu_j^m + \sum_{v_k \in \mathcal{E}(j)} \left(\Sigma_k^{\vec{k}j} \right)^{-1} \mu_k^{\vec{k}j} \end{aligned} \quad (20)$$

where $\Theta_j^m = \{\mu_j^m, \Sigma_j^m\}$ are the parameters of each hyper-sample and $\mu_k^{\vec{k}j}, \Sigma_k^{\vec{k}j}$ the parameters of each message.

A problem with using random samples is that many samples may be required to give confidence that the sampled distribution is accurately represented. One method to provide a confidence in the ability of a sample set to represent the PDF is to measure the KL-divergence between the covariance and mean of the samples and that of the original distribution. The closer to zero this measure

is, the more confidence we have. Instead of this, Eqn. (17), used to select
625 a set of Sigma points, provides a means to deterministically select a set of
delta-samples that exactly represent the covariance and mean of the original
distribution, ensuring the KL-divergence between them is zero. We therefore
sample from the proposal distribution by decomposing it into a set of sigma
points using (17), except that a copy of the mean is also maintained. So, $2D + 1$
630 sigma points are selected and each scaled by $\sqrt{(D + 1/2)v_d}$. An example of the
delta-samples used to represent a single hyper-sample is shown in Figure 10,
projected into two different camera views. The benefit of this approach is that
it requires just 7 delta-samples to be extracted for each node of each hyper-
sample. This makes the approach extremely efficient, since the bottleneck in
635 pose estimation is typically the evaluation of the observational likelihood. Each
delta sample can then be weighted by its likelihood and a Gaussian fitted using
the ML estimate.

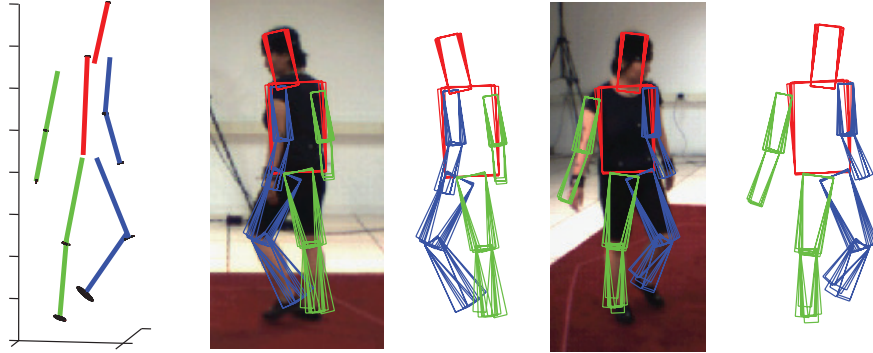


Figure 10: An example of a set of sample points used to estimate observational likelihood distributions projected into two views. They represent the distributions shown on the left.

A message is constructed in a similar way to the belief, except that a message is not received from the node to which the message is being passed. If we define this distribution as $\mathcal{N}(\mathbf{x}_j; \mu_j^{\vec{j}i}, \Sigma_j^{\vec{j}i})$, we can construct a message by propagating this distribution through the limb conditional $p(\mathbf{x}_{ij}|\theta_{ij})$, so that the message is a distribution over \mathbf{x}_i . This is performed using the Unscented Transform by propagating $\mathcal{N}(\mathbf{x}_i; \mu_i^{\vec{i}j}, \Sigma_i^{\vec{i}j})$ through the rotation defined by the mean of the

limb conditional. Hence,

$$\mathcal{N}(\mathbf{x}_i; \mu_i^{ji}, \Sigma_i^{ji}) = \mathcal{F}\left(\mathcal{N}(\mathbf{x}_j; \mu_j^{\vec{ji}}, \Sigma_j^{\vec{ji}}), \mu_{ij}\right). \quad (21)$$

Whilst this propagates the uncertainty in the initial message, the uncertainty in the limb conditional Σ_{ij} must also be passed. This is achieved again using the Unscented Transform and propagating the limb conditional through the mean of the initial message $\mu_j^{\vec{ji}}$,

$$\mathcal{N}(\mathbf{x}_i; \mu_i^{mod}, \Sigma_i^{mod}) = \mathcal{F}\left(\mu_j^{\vec{ji}}, \mathcal{N}(\mathbf{x}_{ji}; \mu_{ji}, \Sigma_{ji})\right). \quad (22)$$

The final message is then given by the convolution of the two of these distributions, setting $\mu_j^{msg} := 0$. The marginal for a given root node state is approximated as

$$p(\mathbf{x}_r^m | \mathbf{Z}) \approx \prod_{i=1}^{n-1} \sum_{s=1}^7 \pi_i^s, \quad (23)$$

where π_i^s is the weight of the delta-sample drawn for the i th part. A hyper-sample S^m then consists of a root node state, a set of updated Gaussian distributions and a weight. The Maximum A Posterior (MAP) pose \mathbf{X}^{MAP} is given by the set of Gaussian centers of the hyper-sample with the highest weight, $\mathbf{X}^{MAP} = \{\mathbf{x}_r^{m^*}, \mu_1^{m^*}, \dots, \mu_{n-1}^{m^*}\}$, where $m^* = \operatorname{argmax}_m p(\mathbf{x}_r^m | \mathbf{Z})$.

Once each hyper-sample has been updated they are then propagated through the temporal priors described in Section 5.1. The new distribution then acts as a prior for the following frame.

5.2.3. Likelihood Function

The observational likelihood used for tracking is based on the binary silhouette. Given a silhouette \mathcal{B} and the set of image pixels \mathcal{P} , pixels classified as the foreground are set to one $\mathcal{B}(\mathcal{P}_{fg}) := 1$ and those classified as the background are set to zero $\mathcal{B}(\mathcal{P}_{bg}) := 0$. The appearance of a part is dependent on \mathbf{x}_i^s , since this will cause changes in scale due to depth or foreshortening due to orientation. The projection of the part consists of the pixels $\mathcal{L}(\mathbf{x}_i^s) \subset \mathcal{P}$ and the cost is defined as $p(\mathbf{z}_i | \mathbf{x}_i^s) \propto \sum_{l \in \mathcal{L}(\mathbf{x}_i^s)} \mathcal{B}(l)$.

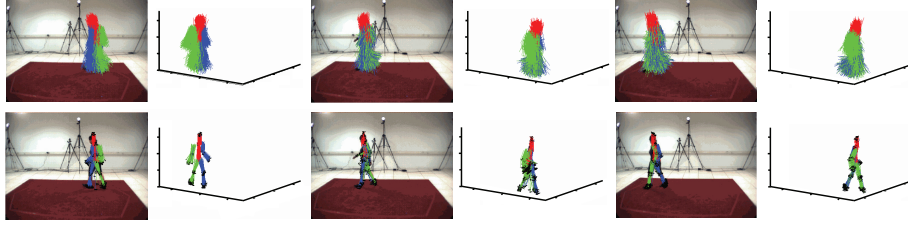


Figure 11: Example frames showing the distribution of the samples using the SIR-PF (top) and the proposed method (bottom). The covariances for each sample have also been plotted for the proposed method.

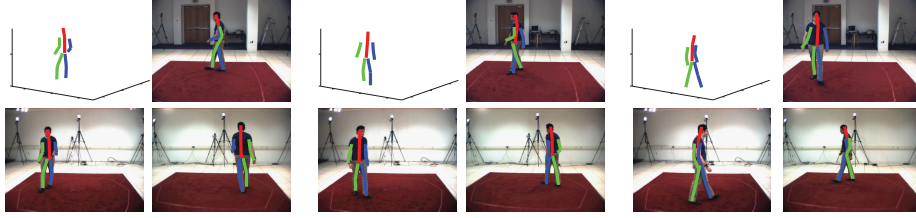


Figure 12: Example frames showing the MAP 3D pose using the proposed method projected into each camera view.

To prevent different limbs being assigned to the same mode (over counting), each constructs a version of the binary silhouette for the opposing part $\mathcal{B}_{opp(i)}$, given by

$$\mathcal{B}_{opp(i)}(\mathcal{L}([\mathbf{x}_i^s]_{s \in \mathcal{D}}) \cap \mathcal{P}_{fg}) := 0.5, \quad (24)$$

This makes it preferable for a limb to be located where the opposing limb is not
655 predicted to be, whilst preferring this over locating a limb to a region of the
image classified as the background.

5.3. Experiments and Results

The presented method was tested using the HumanEva dataset. The “Train”
partition of walking and jogging, consisting of only motion capture data, was
660 used to learn all model parameters. The first 300 frames of the “Validation”

partition was used for testing. Foreground/background segmentation was performed using the Matlab code provided with the data set using default settings.

The presented approach was tested against two existing methods, the Annealed Particle Filter (APF) and the Sequential Importance Resampling Particle Filter (SIR-PF). All methods use the same model parameters, however, whilst the presented method adds temporal diffusion by directly inflating the covariance of each part the alternative methods perform this step stochastically. The APF allows the presented method to be tested against an approach that converges to a single mode. Whilst the SIR-PF can be used to examine how existing approaches behave when permitted to support a larger area of the posterior. This is controlled by adjusting the particle survival rate. To make sure the APF converge to a single mode we use a survival rate per frame of 0.03. To allow both the SIR-PF and the presented method to support a larger area of the posterior, we use a survival rate per frame of 0.93, whilst tracking from video captured at 60Hz.

To ensure the computational cost of each method is the same, all methods use the same number of image likelihood evaluations. The APF uses 5 layers of 160 particles and the SIR-PF use a single layer of 800 particles. The presented method used 114 hyper-samples, since calculating the posterior for each hyper-sample requires the equivalent image likelihood evaluations as 7 SIR-PF/APF particles.

For the APF, pose was estimated using the expectation value of the samples and for the SIR-PF and the proposed method the MAP estimate was used. Limb limits were learned from the training data and used to discard unlikely poses for all methods.

In Figure 11, the set of particles shown represent the posterior for the proposed method and the SIR-PF. As can be seen if the SIR-PF is used to represent a large uncertainty, this uncertainty is present in all parts of the model. This is in contrast to the proposed method where the posterior for each part is updated conditioned on the root node value of the particle, allowing the uncertainty in these parts to remain small. This allows a large region of the root node state to

Table 3: Pose estimation errors (mm) for different methods using varying frame rates and number of cameras.

Frame Rate (Hz)	No. Camera	APF	SIR-PF	Proposed
60	3	118.9 ± 65.5	102.5 ± 8.4	93.7 ± 5.8
60	2	146.2 ± 47.2	110.4 ± 8.9	103.8 ± 12.8
30	3	109.0 ± 27.3	104.8 ± 10.4	97.6 ± 12.2
20	3	120.9 ± 29.6	106.4 ± 10.8	97.2 ± 8.6
15	3	150.0 ± 70.8	114.1 ± 6.6	104.7 ± 9.2

be supported without increasing the uncertainty of the remaining parts. Example frames showing the estimated pose using the presented method are shown in Figure 12, as can be seen the estimated pose closely resembles that of the subject in each frame.

In Table 3, the error is shown for each method averaged over all subjects. As can be seen the proposed method outperforms both the APF and the SIR-PF. We noted that often the APF would fail due to segmentation artifacts that caused the correct mode to be lost. To further illustrate the robustness of the presented method we reiterate the method for two cameras setting. Fewer camera views will result in more ambiguous observations and in these circumstances it will be beneficial to support the posterior over a larger area of the state space until these ambiguities can be resolved.

We further experimented using three cameras but at different frame rates. For all frame rates the annealing rate is adjusted for the SIR-PF and presented method to maintain $\alpha_c = 0.01$, as described in Section 5.1. The annealing for the APF is unchanged to ensure it converges to a single mode. At lower frame rates, when there is greater movement by the subject across consecutive frames, the APF becomes more prone to tracking failure and the presented method continues to outperform both techniques across all frame rates, highlighting its superiority. In Figure 13 we show some example frames of the MAP pose and the distribution of samples used to represent the posterior, whilst tracking at 30Hz.

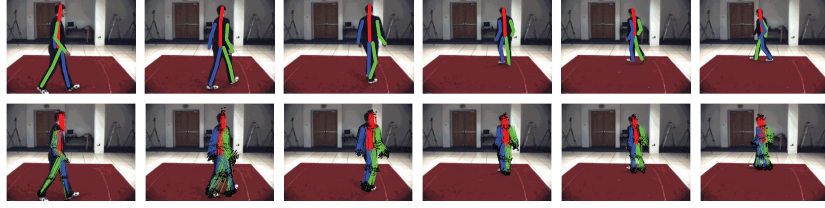


Figure 13: Example frames showing the MAP estimate of pose (top) and distribution of the samples used to represent the posterior (bottom) whilst tracking at 30Hz.

Whilst in some instances the quantitative errors between the proposed method
 715 and the SIR-PF are relatively close, qualitatively the tracking is significantly
 poorer for the SIR-PF. In Figure 14, example frames are shown comparing the
 MAP solution using the SIR-PF compared to the proposed method. As can be
 seen the poses estimated by the SIR-PF are notably worse than those estimated
 by the proposed method. We observed that in general unrecoverable tracking
 720 failure for the APF resulted from poorly estimating the state of the root node,
 for example by estimating the incorrect orientation. This observation highlights
 the importance of representing greater uncertainty over the root node to develop
 robust tracking algorithms for articulated objects.

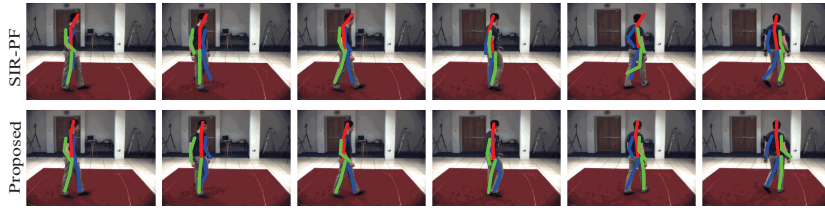


Figure 14: Comparison of pose estimation between the SIR-PF (top row) and proposed method (bottom row).

Very recently comparison results can be found in [49], Simo-Serra *et al.* [50]
 725 proposed to stochastically propagate the noise from the image plane to generate
 a set of ambiguous 3D shapes in the shape space, which is then optimized
 by imposing kinematic constraints, in order to tackle noisy observations. Our
 proposed method achieved better quantitative results in terms of error on the

HumanEva walking dataset, i.e. an average of 103.8 vs. 111.8. Wang *et al.* [49] and Taylor *et al.* [51] showed that the improvements can be achieved by using some strong prior. Wang *et al.* [49] modeled the 3D pose as a linear combination of a set of pose bases, and introduced the constraints on the pose model, including sparsity constraint on the basis coefficients, and anthropomorphic constraint. To recover the 3D pose, they iteratively solve two optimization problems, which are first estimating the bases' coefficient by minimizing the project of 3D pose hypothesis and the 2D pose with respect to the camera parameters, and then re-estimating the pose with respect to the constraints of pose model. Taylor *et al.* [51] introduced Conditional Restricted Boltzmann Machines (CRBM) to model the motion of the subjects, which shows that with the motion prior the error of estimating 3D pose can be reduced significantly. Our tracking approach has no assumed motion model, however, ours could be combined with a much stronger prior, such as [51], to achieve better results.

In both pose estimation and tracking, the bottleneck of computational cost is in computing the likelihood functions that describe how likely it is a part in the given configuration given the observations. In 3D monocular pose estimation if we use a ground plane of $2m^2$ at a resolution of one hyper sample per 100mm with 16 orientations and 600 delta samples per hyper sample we compute 3.8 million likelihood computations. If we were to apply a standard dynamic programming approach and discretized the entire space, over a five part model, where each part has 6 degrees of freedom and each dimension is discretized into 10 bins, which would be extremely coarse we would require approximately 2×10^{12} image likelihood evaluations, which is intractable for any existed method. So, we use the same number of image likelihood evaluations as the competing methods but achieve much better results.

6. Conclusions

In this paper we have presented two novel solutions to extract 3D human pose. The first was from a single monocular image and the second was applied to

multi-view tracking. Both solutions were designed to exploit the key assumption that it is easier to estimate pose if the root node state is known *a priori*. This was achieved by extracting a set of local solutions through the use of hyper-samples. There are two key benefits to this approach that we have exposed. The first is that using a fixed root node allows the human body to be modeled as a kinematic chain that can more efficiently be optimized than alternative representations. The second is that the presented approach allows more of the posterior to be supported than current methods allow. By exploiting the first benefit we have shown it is possible to extract an entire set of solutions using the same computational cost as competing methods would require to find a single solution. The second benefit has been used to engineer a tracking method that is robust in the presence of noisy, ambiguous observations, and to design a single image monocular solution that is not dependent on initialization.

For tracking it was shown that more robust performance can be achieved by providing greater support over the state of the root node. This was particularly emphasized at lower frame rates, where noise and missing data becomes much more detrimental as the weakness of the simple temporal prior becomes exposed. The philosophy of this approach is far removed from the most common to assume the answer lies in strengthening the temporal prior, we believe the solution lies in strengthening the support over the posterior distribution until stronger, more informative observations become available.

References

- [1] Y. Yang, D. Ramanan, Articulated pose estimation with flexible mixtures-of-parts, in: CVPR, IEEE, 2011, pp. 1385–1392.
- [2] M. Andriluka, S. Roth, B. Schiele, Discriminative appearance models for pictorial structures, International journal of computer vision 99 (3) (2012) 259–280.
- [3] S. Johnson, M. Everingham, Learning effective human pose estimation from inaccurate annotation, in: CVPR, IEEE, 2011, pp. 1465–1472.

- [4] J. Deutscher, I. Reid, Articulated body motion capture by stochastic search, *IJCV* 61 (2005) 185–205.
- [5] D. Ramanan, D. A. Forsyth, A. Zisserman, Tracking people by learning their appearance, *T-PAMI* 29 (1) (2007) 65–81.
- [6] R. Urtasun, D. J. Fleet, P. Fua, 3D people tracking with gaussian process dynamical models, in: *CVPR*, 2006, pp. 238–245.
- [7] R. Li, T.-P. Tian, S. Sclaroff, M.-H. Yang, 3D human motion tracking with a coordinated mixture of factor analyzers, in: *IJCV*, 2010, pp. 170–190.
- [8] A. Baak, B. Rosenhahn, M. Muller, H.-P. Seidel, Stabilizing motion tracking using retrieved motion priors, in: *ICCV*, 2009, pp. 1428–1435.
- [9] P. Felzenszwalb, D. Huttenlocher, Pictorial structures for object recognition, *IJCV* 61 (2005) 55–79.
- [10] X. Lan, D. P. Huttenlocher, Beyond trees: Common-factor models for 2D human pose recovery, in: *ICCV*, 2005, pp. 470–477.
- [11] Y. Yang, D. Ramanan, Articulated pose estimation using flexible mixtures of parts, in: *CVPR*, 2011, pp. 1385–1392.
- [12] S. Johnson, M. Everingham, Clustered pose and nonlinear appearance models for human pose estimation, in: *BMVC*, 2010.
- [13] L. Sigal, S. Bhatia, S. Roth, M. Black, M. Isard, Tracking loose-limbed people, in: *CVPR*, 2004, pp. 421–428.
- [14] A. Fossati, M. Dimitrijevic, V. Lepetit, P. Fua, Bridging the gap between detection and tracking for 3D monocular video-based motion capture, in: *CVPR*, 2007, pp. 1–8.
- [15] B. Sapp, D. Weiss, B. Taskar, Parsing human motion with stretchable models, in: *CVPR*, 2011, pp. 1281–1288.

- [16] M. Andriluka, S. Roth, B. Schiele, Monocular 3D pose estimation and tracking by detection, in: CVPR, 2010, pp. 623–630.
- [17] X. Xu, B. Li, Exploiting motion correlations in 3-d articulated human motion tracking, T-IP 18 (6) (2009) 1292–1303.
- [18] T.-H. Yu, T.-K. Kim, R. Cipolla, Unconstrained monocular 3D human pose estimation by action detection and cross-modality regression forest, in: CVPR, 2013, pp. 3642–3649.
- [19] A. Yao, J. Gall, L. Gool, Coupled action recognition and pose estimation from multiple views, IJCV 100 (1) (2012) 16–37.
- [20] M. A. Fischler, R. A. Elslager, The representation and matching of pictorial structures, in: IEEE Trans. Computer, 22(1), 1973, pp. 67–92.
- [21] L. Pishchulin, A. Jain, M. Andriluka, T. Thormahlen, B. Schiele, Articulated people detection and pose estimation: Reshaping the future, in: CVPR, 2012, pp. 3178–3185.
- [22] V. Ferrari, M. Marin-Jimenez, A. Zisserman, Progressive search space reduction for human pose estimation, in: CVPR, 2008, pp. 1–8.
- [23] A. Toshev, C. Szegedy, Deeppose: Human pose estimation via deep neural networks, in: CVPR, IEEE, 2014, pp. 1653–1660.
- [24] M. W. Lee, S. Member, I. Cohen, A model-based approach for estimating human 3D poses in static images, T-PAMI 28 (2006) 905–916.
- [25] J. McCormick, M. Isard, Partitioned sampling, articulated objects, and interface-quality hand tracking, in: ECCV, 2000, pp. 3–9.
- [26] G. Hua, Y. Wu, Variational maximum a posteriori by annealed mean field analysis, T-PAMI 27 (11) (2005) 1747–1761.
- [27] H. Sidenbladh, M. J. Black, D. J. Fleet, Stochastic Tracking of 3D Human Figures using 2D Image Motion, in: ECCV, 2000, pp. 702–718.

- [28] B. Daubney, X. Xie, Estimating 3D human pose from single images using iterative refinement of the prior, ICPR.
- 840 [29] C. Sminchisescu, B. Triggs, Covariance scaled sampling for monocular 3D body tracking, in: CVPR, 2001, pp. 447–454.
- [30] J. Deutscher, A. Davidson, I. Reid, Automatic partitioning of high dimensional search space associated with articulated body motion capture, in: CVPR, 2001, pp. 669–676.
- 845 [31] B. Daubney, X. Xie, Estimating 3D pose via stochastic search and expectation maximization, in: AMDO, 2010, pp. 67–77.
- [32] M. Isard, Pampas: Real-valued graphical models for computer vision, in: CVPR, 2003, pp. 613–620.
- [33] S. Amin, M. Andriluka, M. Rohrbach, B. Schiele, Multi-view pictorial structures for 3D human pose estimation, in: BMVC, 2013.
- 850 [34] A. Kanaujia, N. Kittens, N. Ramanathan, Part segmentation of visual hull for 3D human pose estimation, in: IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2013, pp. 542–549.
- [35] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in: CVPR, 2011, pp. 1297–1304.
- 855 [36] K.-C. Chan, C.-K. Koh, C. Lee, A 3d-point-cloud feature for human-pose estimation, in: IEEE International Conference on Robotics and Automation, 2013, pp. 1623–1628.
- 860 [37] B. Daubney, X. Xie, Tracking 3D human pose with large root node uncertainty, CVPR (2011) 1321–1328.
- [38] B. Daubney, D. Gibson, N. Campbell, Estimating pose of articulated objects using low-level motion, Computer Vision and Image Understanding 116 (3) (2012) 330 – 346.

- 865 [39] A. Cherian, V. Morellas, N. Papanikolopoulos, Accurate 3D ground plane estimation from a single image, in: IEEE international conference on Robotics and Automation, 2009, pp. 519–525.
- [40] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, 2005, pp. 886–893.
- 870 [41] S. Johnson, M. Everingham, Learning effective human pose estimation from inaccurate annotation, in: CVPR, 2011.
- [42] Q. Zhu, M.-C. Yeh, K.-T. Cheng, S. Avidan, Fast human detection using a cascade of histograms of oriented gradients, CVPR (2006) 1491–1498.
- [43] D. G. Lowe, Distinctive image features from scale-invariant keypoints, IJCV
875 60 (2004) 91–110.
- [44] W. Cohen, Fast effective rule induction, International Conference In Machine Learning (1995) 115–123.
- [45] L. Sigal, A. Balan, M. Black, Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human
880 motion, IJCV 87 (2009) 4–27.
- [46] C. Conaire, N. O’Connor, A. Smeaton, Detector adaptation by maximising agreement between independent data sources, in: CVPR, 2007, pp. 1–6.
- [47] S. Johnson, M. Everingham, Clustered pose and nonlinear appearance models for human pose estimation, in: BMVC, 2010.
- 885 [48] S. Julier, J. Uhlmann, H. Durrant-Whyte, A new approach for filtering nonlinear systems, in: American Control Conference, Vol. 3, 1995, pp. 1628–1632.
- [49] C. Wang, Y. Wang, Z. Lin, A. L. Yuille, W. Gao, Robust estimation of 3d human poses from a single image, in: CVPR, IEEE, 2014, pp. 2369–2376.

- 890 [50] E. Simo-Serra, A. Ramisa, G. Alenya, C. Torras, F. Moreno-Noguer, Single
image 3D human pose estimation from noisy observations, in: CVPR, 2012,
pp. 2673–2680.
- [51] G. W. Taylor, L. Sigal, D. J. Fleet, G. E. Hinton, Dynamical binary latent
variable models for 3d human pose tracking, in: CVPR, IEEE, 2010, pp.
895 631–638.

Cronfa - Swansea University Open Access Repository

This is an author produced version of a paper published in :
Image and Vision Computing

Cronfa URL for this paper:
<http://cronfa.swan.ac.uk/Record/cronfa28339>

Paper:

Daubney, B., Deng, J. & Xie, X. (2016). Fixing the Root Node: Efficient Tracking and Detection of 3D Human Pose through Local Solutions. *Image and Vision Computing*, 52, 73-87.

<http://dx.doi.org/10.1016/j.imavis.2016.05.010>

This article is brought to you by Swansea University. Any person downloading material is agreeing to abide by the terms of the repository licence. Authors are personally responsible for adhering to publisher restrictions or conditions. When uploading content they are required to comply with their publisher agreement and the SHERPA RoMEO database to judge whether or not it is copyright safe to add this version of the paper to this repository.

<http://www.swansea.ac.uk/iss/researchsupport/cronfa-support/>